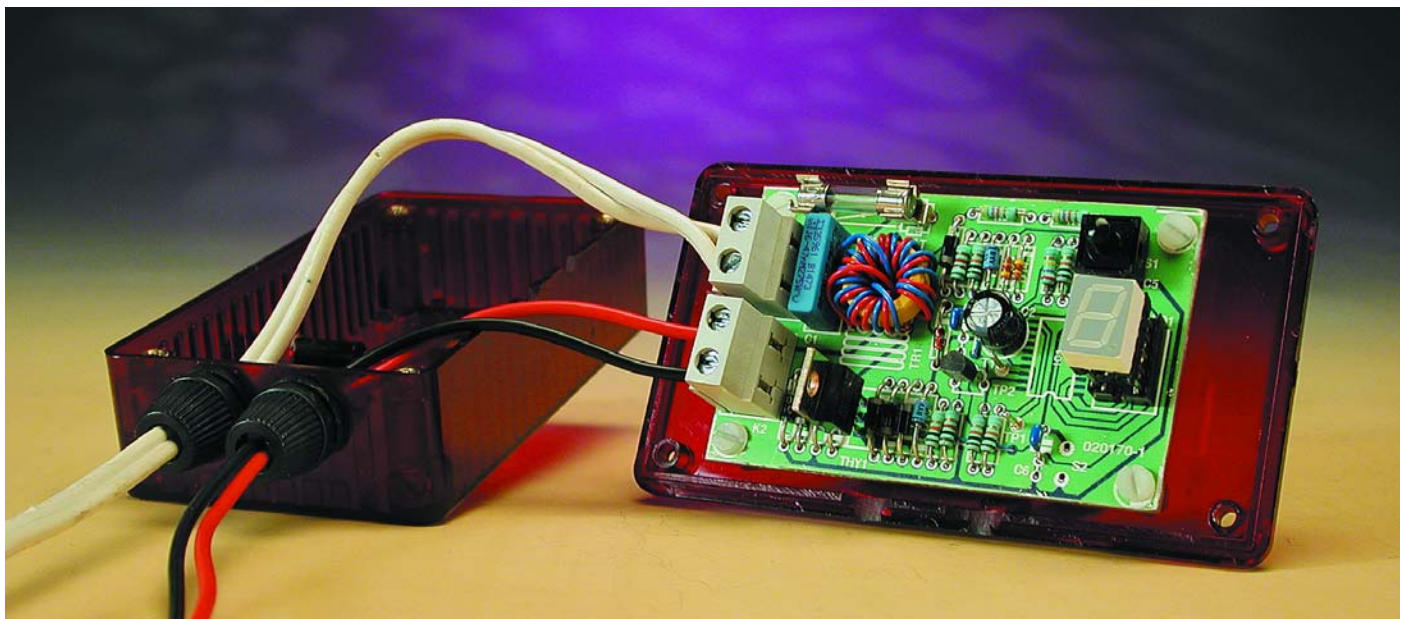


Intelligent Fan Timer

Using the MSP430 low-power processor

Design by J. Volkering

This processor-driven automatic fan controller does not just run in parallel with the toilet light, but instead allows the user to adjust the running time of the fan using a single-button system. The selected time can be read from a single-digit display. The fan can also be stopped early using the same button.



Most toilet fans are switched on and off along with the toilet light, and some continue to run for a certain amount of time after the light is switched off. Since the need for fresh air is closely related to the particular reason for visiting the toilet, the author of this article wanted to have a bit more influence over how long the fan continued to run. The nicest solution would be to have a fan whose running time could be entirely chosen by the user. Naturally, switching the fan off early should also be possible, and it would be a useful extra feature to have the operating time of the

fan shown on a display.

In itself, such a circuit is not particularly difficult to implement. However, when it is necessary to count down a preset time, control a display and drive a switching element, using a small microcontroller can easily prove to offer advantages. This also makes it possible to keep the operating controls simple. Here we have chosen a single-button system, which works very nicely in practice: you press the button once to switch

on the fan, press it repeatedly to extend the time, and hold the button pressed to switch off the fan.

If a circuit of this sort is to be made as compact as possible, it is helpful if it can be powered directly from the mains without a transformer, using a simple voltage stabiliser, rectifier and series resistor. To avoid dissipating too much power in the series resistor, the circuit should not draw too much current, which means we need to look for a low-



power processor. The Texas Instruments MSP430 series meets these economy requirements, and it has several other features that allow the number of components needed to be kept to an absolute minimum.

The Texas Instruments MSP430 family is a family of 16-bit RISC micro-controllers especially developed for low-power applications. Over the course of time, the family has been extended with versions having a mix of functions, such as A/D converters, comparators, timers, PWM outputs, watchdogs, LCD controllers, UARTs, synchronous serial ports and a multiplier. The on-chip memory can consist of a combination of RAM, ROM, one-time programmable (OTP) ROM, EPROM and Flash memory.

voltage ranges from 1.2 V to 5 V, depending on the type.

All members of the MSP430 family have an extensive oscillator circuit on board, with the choice of an internal oscillator, an inexpensive 32-kHz watch crystal or a crystal for the desired clock rate (8 MHz maximum). The frequency of the internal oscillator or digitally controlled oscillator (DCO) is adjustable, thus allowing the 50-Hz mains frequency to be used as a reference.

For the fan timer, we chose one of the least expensive types, the MSP430F1121. It has only 1 kbyte of Flash memory, 128 bytes of RAM, a comparator, a 16-bit timer with PWM outputs and a watchdog timer.

Most of the signal pins of the MSP430F1121 can be programmed as digital I/O pins according to the application, and each input can independently generate an interrupt. If the I/O ports are configured as outputs, the total output current can be as much as 48 mA. Other functions of the MSP430F1121, such as the comparator and the PWM outputs, can also be programmed to be connected to the output pins.

cuit board. The MSP430 family is programmed via a serial port, which is called a JTAG port, and the software is also tested using the same port. This port is shared with the general I/O pins.

More information and sample programs can be obtained from the Texas Instrument website at

[http://focus.ti.com/docs/prod/folders/
print/msp430f1121.html](http://focus.ti.com/docs/prod/folders/print/msp430f1121.html)

Figure 1 shows the complete schematic diagram of the fan timer. As can be seen, it does not involve very much hardware.

The supply voltage is taken directly from the mains network via connector K1, fuse F1 and a noise-suppression choke (Tr1). Tr1 is wound bifilar in order to present a high impedance to noise pulses, which are typically asymmetric, while C1 is specifically included to provide a short circuit for undesired noise pulses.

After half-wave rectification by D1, the supply voltage is stabilised at a nominal level of 3 V using the series combination of R1, R2 and D2. The series resistance for the Zener diode has been intentionally been divided between two resistors in order to keep the voltage drop across each resistor within limits and allow standard 0.5-W resistors to be used. The average current through this series circuit is approximately 5 mA. The combined average dissipation of R1 and R2 is thus 0.5

W, which is the power consumption of the timer in the quiescent state.

Capacitor C2 has a relatively large value, since it must smooth the ripple in the supply voltage resulting from half-wave rectification as well as provide an energy buffer to allow the decimal point of the display to be illuminated every five seconds when the timer is quiescent.

The 50-Hz reference signal is generated by R3, R4, D3, D4, D14 and C4. Here again, the series resistance is divided between two resistors to limit the voltage across each resistor. Schottky diode D3 has been added to prevent the 50-Hz reference signal from going more than 250 mV below the ground level of the supply, in order to prevent damage to the MSP430. C4 suppresses any high-frequency noise that may be present at the 50-Hz reference input.

The load (the fan motor) is switched on and off by a thyristor (THY1) incorporated in a diode bridge formed by D5–D8. In effect, the thyristor shorts the plus and minus terminals of the bridge so that an alternating current can flow through the bridge.

When the thyristor is conducting, a portion of the full-wave rectified sine wave is present across the bridge, with a peak voltage of approximately 1.8 V. The supply voltage for the display is taken from this voltage. The advantage of this is that the display draws energy only when necessary, thus avoiding placing an extra load on the supply for the MSP430. Schottky diode D13 has also been added here to prevent the pins of the microcontroller from going more than 250 mV below the supply voltage.

The segments of the display can be energised by pulling the associated outputs P1.0–P1.7 low. The output resistance of the microcontroller ports, which is around 100 Ω , limits the current through the display segments to approximately 3 mA. However, the LED display should have a voltage drop of at least 1.5V across the LEDs, since otherwise external current-limiting resistors must be added. This was not necessary with the type shown in the components list. In order to obtain sufficient light intensity, it is recommended to choose a display with high optical efficiency at low currents. The selected type (HD1105-O) provides 21,000 $\mu\text{candela}$ at 10 mA.

At the zero crossing of the mains voltage, the thyristor will go off when there is no anode–cathode voltage and no gate voltage. However, if the gate voltage is held at 3V, it will immediately switch on again as soon as the voltage across the thyristor is greater than the peak on-state voltage (V_{tm}) of 1.7 V. This avoids switching spikes during the

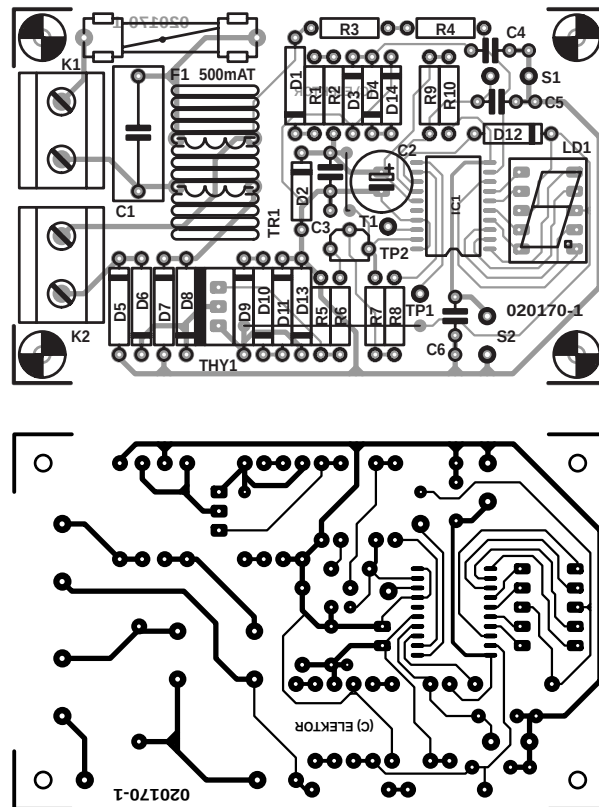


Figure 2. The track and component layouts for the timer printed circuit board.

active state. A supplementary benefit is that there is no need for special timing software to synchronise the gate drive to the 50-Hz signal, except when it is first switched on.

In order to prevent voltage spikes that may be induced at the gate of

the thyristor from disturbing the operation of the MSP430, the thyristor is driven via a transistor buffer stage using T1. This would not have been necessary just to supply the gate current, since the MSP430 has enough 'muscle' to drive the thyristor directly.

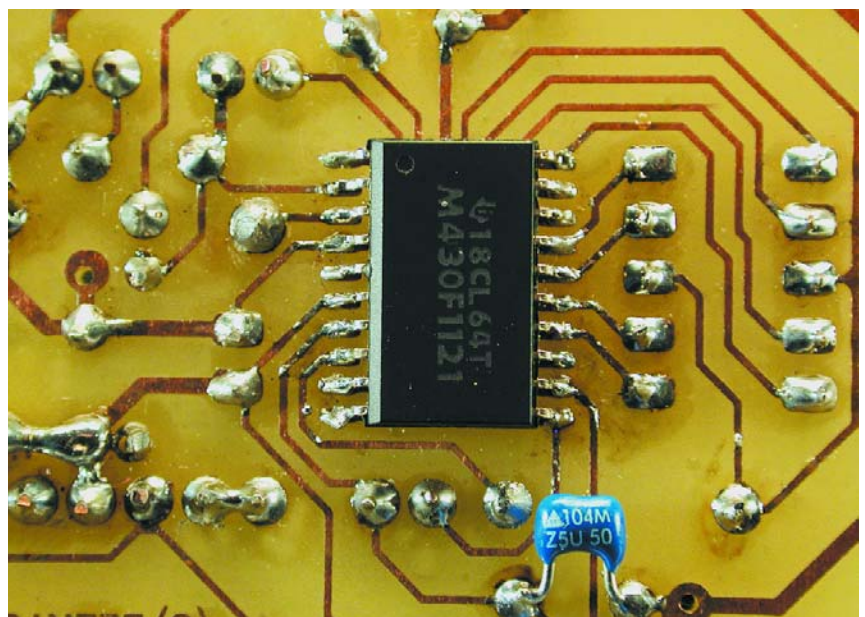


Figure 3. Fitting the MSP430 requires better than average soldering skills.

COMPONENTS LIST

Resistors:

R1 = 8k Ω
 R2, R5, R7, R8, R10 = 10k Ω
 R3, R4 = 47k Ω
 R6 = 1k Ω
 R9 = 270k Ω

Capacitors:

C1 = 47nF 400V, Class X2
 C2 = 220 μ F 16V radial
 C3-C6 = 100nF

Semiconductors:

D1, D4-D11, D14 = 1N4007
 D2 = zener 3 V/500 mW
 D3, D13 = BYV10
 D12 = 1N4148
 T1 = BC327
 THY1 = TIC106D
 LD1 = HD1105-O
 IC1 = MSP430F1121, programmed,
 order code **020170-41**

Miscellaneous:

K1, K2 = 2-way PCB terminal block,
 lead pitch 7.5mm
 F1 = fuse, 500 mAT (time lag), with
 PCB mount holder
 S1 = pushbutton, 1 make contact,
 Class 2 (for test purposes only)
 S2 = pushbutton, 1 make contact,
 Class 2, e.g., Omron type A3DT-
 7111 with cap A3DT-500R (Farnell)
 Tr1 = approx. 20 turns, bifilar, on
 Philips core TN14/9/5 (material:
 3C85)
 Enclosure: e.g., Hammond type
 1591B (red)

Disk, project software, order code
020170-11 or Free Download

PCB available from **The PCBShop**

The MSP430 hardware

The display is connected to port P1.x, with the segments being illuminated with the associated outputs are low.

A reset circuit (R9, C5 and S1) has been added so that the microcontroller starts up in a well-defined state on power-up. The reset switch is only intended to be used for testing. Once the circuit has been fitted

into an enclosure, switch S1 will anyhow no longer be accessible.

Pushbutton switch S2 is used to activate and stop the timer. If an I/O pin of the MSP430 is programmed as an input, the input signal passes through a Schmitt trigger, so a relatively slow input edge does not pose a problem for the MSP430. The contact bounce of the switch can thus be suppressed using a simple capacitor connected to the input, without

requiring debouncing in software.

When the fan is not running, the internal 16-bit timer of the MSP430, which is driven by the internal oscillator, is used to indicate that the circuit is quiescent by energising the decimal point of the display for half a second every five seconds. This is done by setting port P2.0 high to apply a voltage to the anode of the display via D12. The decimal point of the display is then briefly pulled low by P1.0. When the fan is active, port P2.0 is held low to avoid loading the 3-V supply. Power for the display is then taken from D9-D11.

When S2 is pressed, the MSP430 switches from the quiescent state to the operational state. The hardware timer is switched off and the internal comparator is programmed to compare the 50-Hz input signal to 0.25 V_{CC} (750 mV). The output of the comparator is fed to port 2.5 (test point TP5). Every 20 ms, a comparator interrupt is triggered by a zero crossing of the mains voltage.

The entire housekeeping of the timer is derived from this 50-Hz interrupt, while the MSP processor is clocked by the internal oscillator (at approximately 1 MHz). First the display is energised, and on the next zero crossing, the thyristor is triggered into conduction.

The MSP430 software

We can start with the reassuring news that you can purchase a pre-programmed MSP430 chip via Readers Services. For those who like to do things themselves, the software and source code are also available on a diskette, or they can be downloaded from the *Elektor Electronics* website.

Now for a few details about the software. There are four software counters active in the MSP430. The first one runs in the interrupt service routine for the 50-Hz comparator and counts up to 50, in order to generate a 1-Hz signal (test point TP1). This 1-Hz signal in turn serves as a clock pulse for the second software counter, which counts up to 300. This is equivalent to 5 minutes, which is the minimum duration for which the timer can be activated. The 5-minute counter pulse, in turn, drives a software counter whose value matches the indication on the display. If S2 is pressed repeatedly, the displayed value can be increased up to a maximum of 9. This is equivalent to an activation time of 45 minutes. There is no use pressing S2 again after this value is reached, since 9 is the maximum.

If the timer must be switched off early, S2 must be pressed and held for more than three seconds. As soon as S2 is pressed, a fourth counter starts incrementing every second as long as S2 remains pressed. If S2 remains

Programming your own microcontroller

There is an inexpensive development tool available for trying out the MSP430, which is called the Flash Emulation Tool (FET). In practice, this tool also proves to be suitable for developing complete applications, such as the fan timer described in this article.

The FET (MSP-FET430x110; see www.ti.com) is supplied with a limited version of the IAR workbench development environment, called 'Kickstart'. It is limited in terms of the size of C programs accepted by the loader/debugger, which is 2000 lines after compilation. This amount of lines is certainly necessary, since the header file that is also supplied occupies nearly 1000 lines by itself, but it does give you descriptions of all the registers and bit positions of the entire processor. That saves quite a bit of work when you are writing your first program.

The code/debugging limitation does not apply if you work with assembler. For those who like to write in assembler, the MSP430 family is easy to program, since it has an orthogonal processor. This means that all instructions can use all possible types of addressing on all registers and RAM. Since the MAP430 has a RISC architecture, there are only 27 different basic instructions, although there are 7 different addressing options. The MSP430F1121 FET tool is connected directly to the printer port of a PC and costs approximately £35.

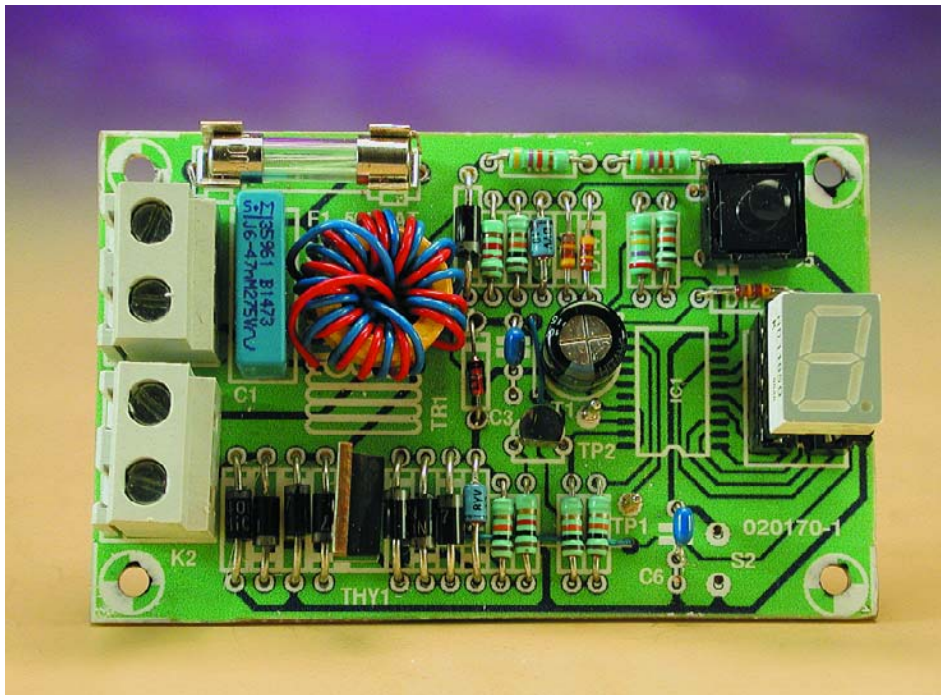


Figure 4. The component side of a fully assembled board.

pressed for longer than 3 seconds, this counter reaches a value of 3 and the thyristor is switched off.

The fan is switched on and off within the 50-Hz interrupt service routine, which is triggered on the zero crossings of the mains voltage. The timer thus always switches the fan on or off on a zero crossing.

When the timer switches off or is not active, the comparator and associated interrupt are disabled, and the internal hardware counter is again enabled in order flash the decimal point of the display to indicate that the circuit is in the quiescent state. The watchdog is continuously addressed during both the quiescent and active states, in order to prevent it from generating a system reset.

Practical implementation

The circuit board layout designed for the fan timer is shown in **Figure 2**. This circuit board is available from the PCB Shop (see the *Elektor Electronics* website).

There is only one difficulty with building the circuit board, which is fitting the MSP430 SMD IC. This little jewel is fitted to the solder side of the board, and it is a very good idea to do this right at the start.

First tin the solder pads on the board, and then clean them a bit using desoldering braid. Place the IC on the solder pads, and then attach two corner pins using a small soldering iron with a sharp point. After this, solder the remaining pins. Be frugal with the sol-

der, in order to avoid shorts between the pins. Check to make sure that no undesired solder bridges have been formed and that all pins are securely soldered. Excessive solder can be removed using desoldering braid. A fitted SMD IC is shown in **Figure 3**.

After this, you can turn over the board and fit the 'normal' (leaded) components to the board. This is best done in the usual manner, working from low to high, starting with the wire link next to C3 and finishing with the relatively large circuit-board terminal strips K1 and K2.

The noise suppression choke Tr1 can easily be wound on a small (14 mm diameter) toroidal core (Philips type TN14/9/5 with 3C85 material). Take two lengths of insulated installation wire with a diameter of approximately 0.5 mm and wind them around the core approximately 20 times. The ends of the wire at the beginning of the windings are terminals 1 and 3 in the schematic diagram, while the other ends are terminals 2 and 4.

The display should be mounted on a socket, possibly elevated somewhat by using two 4-way pin headers. Pay careful attention to the polarity of the diodes, since a mistake can result in damage to the microcontroller. The component side

of a correctly build board is shown in **Figure 4**.

After all components have been soldered in place, a seal coat should be applied to the solder side. An inexpensive way to do this is to use transparent nail polish (without glitter!).

After this, the board can be fitted into an enclosure. With regard to electrical safety, only plastic enclosures are suitable. The prototype was fitted into a transparent red plastic Hammond enclosure. This has the advantage that there is no need to saw an opening for the display, since the displayed number can be read through the plastic. If you do make an opening in the enclosure for the display, it must be covered by a piece of transparent plastic for safety. With a transparent enclosure, the only openings needed are a hole for the connection wires and a mounting hole for pushbutton switch S2. Be sure to use the switch shown in the components list for S2, and use plastic screws to attach the circuit board to the enclosure.

When testing and installing the circuit, remember that it is electrically connected to the mains network! Always disconnect the mains leads from K1 before working on the board, and before you start, read the Safety Information page published from time to time in *Elektor Electronics*. If you want to do things fully by the book, attach an identification label to the bottom of the enclosure once you are done, showing the project number and the value of the fuse.

Conclusion

The amount of power this circuit can switch depends on the diode bridge (D5–D8) and the thyristor (THY1). The types shown in the components list can handle 1 A. If you want to switch larger currents, you will have to use correspondingly higher-power components. Note that the thyristor must be a type that triggers with a minimum gate voltage of 2 V at a gate current of less than 100 μ A, since the drive current for the thyristor is provided by the 3-V supply.

(020170-1)